LISTINGS NEWSLETTER

Newsletter of the
Long Island Sinclair/Timex
Users Group

Issue: OCTOBER 1996

ZX81
TS2068
QL

Please send all inquiries and submissions (including dues) to: L.I.S.T.
Mr. Harvey Rait
5 Peri Lane,
Valley Stream, N. Y. 11581

COMING EVENTS: The next L.I.S.T. meeting will be Sunday, 11/17/96 at 2 P.M. at the home of Harvey Rait (see address above).

ZX81
TS2068
QL

TIPS

TIPS

Listing Policy *Annual Dues $16.00*

One 'sample' copy sent upon receipt of Business size SASE. Copies provided on EXCHANGE BASIS with other bona fide user groups. LISTing is published monthly except July and August by LIST (Long Island Sinclair Timex) Group, a not for profit user group We are always looking for articles, programs, reviews etc. to keep our members informed and entertained. You maintain full credit and copyright. Portions of this publication may be reproduced, without written consent. Please give credit to LIST when reprinting articles. LIST disclaims responsibility for any damage you may do to your computer as a result of reading any articles in LISTing.

QL CORNER

Last month in QL Corner, it was reported that a QL user in the UK had operated his QXL IBM emulator was running at a very fast rate. Stuart Honeyball stated in Issue three of QL Today, page 52, "80 Mhz QXL Caution!", stated that the only safe bet for QXL longeviety when upgrading for more speed, was to use a 50 Mhz crystal Oscillator in place of the original 40 Mhz oscillator. Replacing the QXL with a crystal oscillator above 50 Mhz will surely reduce the life expectancy of the processors. Stuart stated that unfortunately, he did not have any numbers on the problem, but it can be a real problem. The degree of the problem depends on the case temperature of the processor more than the clock speed.

By increasing the clock speed, the processor uses more energy and therefore generates more heat. By attaching a fan and supplying proper system circulation the case temperature can be held down. However, if this is not done properly or the speed is increased so much that the fan cooling method can not compensate, the processor will definitely have a decreased life expectancy.

How far can you push your clock speed depends on the individual processor chip (variance in the manufacturing process affect this), the cooling methods you use, and just what you expect for a life expectancy. The clock ratings on a processor from a manufacturer are tested for a given case temperature and life expectancy.

With proper care, it should be feasible to push some chips from 20 to 25 Mhz with probably negligible loss of chip life. However, going beyond that will most likely have an impact and should be avoided if possible. Instead, it is better to buy a higher clock rated processor (don't forget to also improve your cooling methods).

A final note is that other chips in your system that respond to the clock increase (such as the PAL) also have speed limits and can generate additional heat. However, the processor on the QXL will most likely to be the biggest concern when pushing the clock rate.

Dilwyn Jones spoke to Stuart about the above problem and he also expressed some concern about driving the QXL to such extreme speeds. He cited a case where an over-clocked QXL appeared to function normally one day, then refused to work the next day, and the only difference was that the second day was warmer that the first day. It seems clear that pushing the QXL to these extreme speeds is risky, and if the chips blow up, you,ll have only yourself to blame!

QUANTA LIBRARY - UPDATE 09/96

There are nine disks in total, 5 replacement disks, plus 4 new additions: New additions are: PS 07, SP 14, SP 15 & UG16. The library disk LG 01 has been upgraded to include the four new program disks and new files for the remaining five additional disks.

The following is a summary of the changes made to the library, for your information.

Disk No.		Action.
LG 01	*	Updated 09/96
DE 10		Style check demo - Geoff Wicks
GS 10		PE Games - Roy Wood
PS 07		IGItoAR - Roy Kempton

UG 12		Phone Log - Mark Knight Upgrade
UG 13	*	File Utilities - Mark Knight Upgrade
UG 16	*	PE Kit - John Miller
SP 15	*	QL Index - Roy Wood, 'Flashback' version
SP 14		QL Index - Roy Wood, 'DataDesign' version

Please note: The disks with an '*' are disks which have many bad sectors. These disks have been sent back to the Quanta Head Librarian for replacement.

Any Quanta member may request any library disks from this listing and/or any other Quanta library disks. You may want to receive the library disk before ordering disks from the library. Please send me formatted 720K disks with return postage and send to: Bob Gilder, 69 Jefferson Place, Massapequa NY 11758 - Telephone: 516-541-2271

The September issue of Nuts & Volts magazine has a great article on how to repair PC Switching Power Supplies. The writer, T. J. Byers, makes the repair almost painless, with easy reading text, charts, schematics and diagrams. If you are interested in repairing switching power supplies or you just like to familiarize yourself with these units, send for a copy of the September, Nuts & Volts magazine, 430 Princeland Court, Corona, CA 91719-9938. A single issue of the magazine is $3.50. Or, you may be able to find a copy at your news stand.

During the past two years I have purchased approximately six hundred 3.5 inch, 720K diskettes. Some of these purchases were for 'over labeled' diskettes, which are disks that contained software and then re-formatted. I had lots of problems with these disks. Approximately one-third of them had bad sectors and tracks rendering the media useless. Looking around, I have located a good disk supplier for 720K and 1.44K diskettes at a very reasonable price for lots of a hundred. I have purchased 400 of the 720K disks and all were perfect - not a bad track or sectors. The cost for these diskettes are $21.00 per one-hundred. The 1.44 Meg diskettes go for $23.00 per 100. The diskettes do not come with labels, however, they sell 3.5 inch labels at $3.00 per 100. The source for these diskettes: Diskettes Unlimited, 6206 Long Drive, Houston, Texas 77087 - Telephone: 713-643-9939 Fax: 713643-2722.

If you are looking for a source for ED diskettes (3.2 Meg), Midwestern Diskette Computer Supplies sells them in packs of 10, at $1.79 each ($17.90 per pack). They are the only supplier that I have found who carries ED diskettes. Midwestern Diskette, 509 West Taylor, Creston, IA 50801 Tel:800-221-6332 Fax: 515-782-4166.

ED drives can still be purchased at many of the IBM Computer shows for approximately $25.00 each. They are Sony drives and only support 1.44 Meg and 2.88 Meg (they will format 3.2 Meg with a Gold Card or Super Gold Card). I gave my brother one of my ED drives for his IBM computer and he advised me that he can format and write to 720K diskettes (as well on 1.44 Meg and 2.8 Meg) using this drive.

It is my understanding Don Walterman (QBOX) has a hardware jumper patch for for the IBM ED drives allowing all three densities to be formatted. Perhaps Don can share his modification for these drives with QL users.

See you next month... *Bob Gilder*

UPDATE FOR Z80 EMULATOR by Brian Gaff

[Mr Gaff, of England, posted this on COMP.SYS.SINCLAIR on 10 September 1996. He's at briang@bgserv.demon.co.uk. John Pazmino did some spot editing and left info on registering thru just Brian's own B G Services.]

Z80-304.ZIP at an ftp near you now. (PS accept no CD ROM copies!)

Sinclair ZX Spectrum Emulator 'Z80' v3.04 - 10-VIII-96 - by G.A. Lunter

'Z80' is a shareware program. The program is not completely functional, and the parts which are left out are included when you register. You are encouraged to give this demo version to friends, but DO NOT change the original archive in any way, please. The shareware version of the emulator consists of the Z80-304.ZIP archive file, which should contain the following 12 files:

Z80.EXE	- The emulator itself.
Z80.INI	- Default initialisation file (used by Z80.EXE).
Z80.PIF	- Program Info File to run 'Z80' under Windows
Z80.ICO	- Windows icon.
Z80.DOC	- Documentation file for the emulator.
Z80FAQ.DOC	- Frequently asked questions and answers.
REGISTER.DOC	- This file.
NEW.DOC	- The What's New file.
ROMS.BIN	- Various ROM images (used by Z80.EXE).
TECHINFO.DOC	- Technical info about the Spectrum etc.
LAYOUT.SCR	- Keyboard layout help screen (used by Z80.EXE).
DIAGRAM.Z80	- Circuit diagram for tape interface, and calibration

The shareware version of the emulator differs from the registered version in the following respects: it displays a message when the program is started, and will only run normally for five minutes, after which the program cannot be slowed down anymore. Furthermore, it cannot load from or save to tape, and it does not emulate the DISCiPLE and the Plus D interfaces. Everything else works as in the registered version.

If you register, you get the fully working version, and the following utilities:

CONVERT - a general conversion program: can list out BASIC and translate it back, produce .GIF or .PCX files from screen dumps, translate Spectrum ASCII (CR) to PC ASCII (CR/LF), and some other things.

CONVZ80 - Translates various snapshot and tape formats of other Spectrum emulators into each other. Can handle the familiar .SNA format use by several emulators (JPP, XZX,...), and also Pedro Gimeno's (VGASPEC and SPECTRUM) .SP format and Kevin J. Phairs' (SPECEM) .PRG format. It can also handle tape files of SPECEM and L. Rindt and E. Brukner's emulator ZX.

DISCIPLE - Reads DISCiPLE and Plus D diskettes, both 3.5'' and 5.25''. It translates the 48K and 128K snapshot files to .Z80 snapshots, and ordinary files and screen snapshots to .TAP tape files. (Not necessary for the fully registered version, but handy for the cheaper one.)

ADDDAT - Utility to edit the additional data part of .SLT snapshot files, which contain loading screens and level data.

Z802TAP - Converts a .Z80 snapshot, 48K or 128K, to a .TAP file which can be loaded into the emulator and saved to tape by the next utility:

TAP2TAPE - Saves the contents of a .TAP file back to tape, to load it into an ordinary Spectrum.

TAP2VOC - Converts a .TAP file to a .VOC sound sample file, to write to tape, or to load into the emulator.

READVOC - Reads in a long, 'digital' .VOC sound sample, to be used as input to the emulator, from the LPT tape interface or a SoundBlaster.

READSB - Reads .VOC sound samples from SoundBlaster using DMA, and applies a digital filtering and oversampling algorithm tuned for Spectrum tapes.

OUT2VOC - Converts .OUT log files into .VOC or raw sound sample files, so that you can easily extract music samples from Spectrum/Spectrum 128 games, or SAVE directly to a .VOC or raw sample file.

Z80DUMP - Shows the header and the contents of a .Z80 file.

You will also receive the source files of the emulator, the above utilities and the SamRam, and you will be kept informed about future updates.

You can also choose to register for the emulator without DISCiPLE and Plus D emulation, if you are not interested in those parts. The registration fee is a bit lower then, but you will receive everything stated above (including the DISCIPLE program); the only difference is that you get a version of the emulator that will not emulate the M.G.T. interfaces.

There are several registration sites, the main one being B.G. Services in the U.K. You can also register with Jimaz in the Czech Republic, or David Pomeroy in New Zealand, whichever is most convenient. Please note that Friendware in Spain and Sinclair Freakeren are no longer registration sites.

IMPORTANT NOTE:

When sending registrations, please either PRINT your order and address, or use BLOCK CAPITALS to aid readability. If you use a postal cheque service, or direct transfer, please try to also send a letter to the site with details of the payment etc. Many payment slips are unreadable copies.

The registration fee is BP 20 or BP 15 (BP=British Pound) for the version with or without DISCiPLE/Plus D emulation respectively. Payment can be by cheques (Sterling on a UK bank, or Eurocheques) or UK postal order made payable to B G Services. You can also transfer directly to the Giro account 324.82.16 (B G Services). Girobank plc, Bootle, Merseyside G1R 0AA.

You can also send money in cash, in which case also US Dollars and German Marks (registration fee US$ 30/US$ 25, DM 50/DM 40 respectively) can be accepted.

Please do not use other currencies than British Pounds on cheques; it is not economic to change. If it is the only option available, please add BP 8 to cover drawing costs. I regret I cannot accept orders via charge or credit card.

The address is:

B G Services
64 Roebuck Road
Chessington
Surrey KT9 1JX
(United Kingdom)

Telephone enquiries on (0181) 287 4180, Fax (0181) 391 0744, or from abroad: +44 181 287 4180 and +44 181 391 0744 respectively. B G Services can also supply ready built tape interfaces for BP 14.50. These are professional quality items built into 25W 'D' connectors. (If this archive is old, please enquire on price before ordering) Files will be on a 3.5" HD disc together with some ready converted software, which is either PD, or has the approval of the authors. Other disc sizes may be requested, but these may not contain all the extra files.

If you have an older registered version and wish to upgrade, please contact your registration site, sending return postage. Updates within the same version number and normally sent free if a disc and return postage is received. Updates from previous versions will carry a small update fee.

HI-RES GRAPHICS ON THE ZX81 by Steve McDonald.

[Mr McDonald, of England, posted this in COMP.SYS.SINCLAIR on 16 August 1996. He's at stevemcd@easynet.co.uk. Some spot editing by John Pazmino.]

Since I have been receiving many requests for an explanation on how high res is achieved on a standard ZX81 without any additional hardware, I have decided to type up the following enclosed document. Hope it is of some use.

P.S. - Check out the Sinclair ZX81 Support Page at:

http://easyweb.easynet.co.uk/~stevemcd/zx81.htm

Introduction

As a lot of people have been asking me recently, as to how the high res routines actually work, in response, I have decided to detail and release the following code:

(Revision 2.0 - Steven McDonald - August, 1996)

Technical Notes:

A call to HI-RES will start generating a high res display. The HRESGEN routine address is loaded into the IX register and will take over the display output - 50 times a second (UK). A further call to LO-RES will start to generate the normal lo res display again. The Z80 "I" register normally holds the (start of the ZX81 character table)/256 (i.e. 30 * 256 = 7680).

As nearly all of the ROM characters have a blank line at the top, the high res routine changes this to 12 which is at ROM address 3072 which I found contains much more interesting bit-patterns than other values which can be used. Due to hardware limitations, the "I" register can ONLY point into the ROM - and not RAM (hence the reason for only pseudo high res - sigh!).

The high res routine works by fooling the ULA into thinking it is only on the top line of a character - every time! It cleverly does this by resetting and restarting the ULA ROW counter at every VSYNC pulse with the "IN A,(C)" and "OUT (FF),A" code sequence. This allows the same row of patterns to be used for every one of the 192 tv lines.

As this procedure is carried out for every tv line, true high res is achieved vertically - but not horizontally, as this depends on the limited bit patterns stored at the location pointed to by: (the "I" register * 256 + 8 * the value of the "character" stored in the display buffer) Phew! This high res display buffer is stored at (E71E+0021) - 8000 ie. at address 673F - in this example routine.

It consists of 32 decimal display characters or "patterns", followed by a 201 decimal byte which just happens to be the Z80 "RET" opcode. This is repeated 192 decimal times for the complete high res display buffer! In lo res, each line ends with a 118 decimal byte - which is the Z80 opcode, "HALT" - which conveniently waits for the next interrupt!

Basically, any byte with BIT 6 set stored in the display buffer, tells

the ULA to stop outputting tv data and execute that instruction. As it happens "RET" also has BIT 6 set so is executed, thus cleverly returning control to the high res display routine!

These routines work on a real 16K ZX81 and provide pseudo high res displays. The ZX81 emulator, "XTENDER" also follows what is going on, by trapping an immediate change of value to the IX register pair and kicking in its own high res screen handler to generate the correct picture.

Therefore, under the emulator, this high res routine is not actually executed as such - just cleverly emulated!

"XTENDER" first tries to emulate the pseudo high res by looking for the characteristically shaped 6k display buffer in the bottom 16k of RAM and outputting it to the PC screen appropriately.

If it cannot locate this display buffer, normal resolution will be switched back in again.

The Routines

```
HI-RES: HALT            ; Wait for an interrupt to sync the changeover!
```

```
         LD A,(FRAMES)    ; Wait
         LD C,A           ; for the
SYNC1:   LD A,(FRAMES)    ; next
         CP C             ; new
         JR Z SYNC1       ; tv frame - used to sync the changeover!
         LD A,0C          ; Change the value of the "I" Register -
         LD I,A           ; so the top line chosen is more interesting!
         LD IX,HRESGEN    ; Pointer to the replacement high res routine set!
         RET              ; Return

LO-RES:  HALT             ; Wait for an interrupt to sync the changeover!
         LD A,(FRAMES)    ; Wait
         LD C,A           ; for the
SYNC2:   LD A,(FRAMES)    ; next
         CP C             ; new
         JR Z SYNC2       ; tv frame - used to sync the changeover!
         LD A,1E          ; Reset the "I" Register
         LD I,A           ; to the ROM default (30 decimal)
         LD IX,0281       ; Pointer to the ROM display routine set!
         RET              ; Return
         HRESGEN:LD HL,E71E       ; Start address of (HRES DFILE - 33) + 32768
                          ; i.e. Set BIT 15 of address for ULA
         LD DE,0021       ; Amount to add each time for address of next line
         DI               ; Disable interrupts
         LD C,FE          ; The ULA port address
         LD B,16          ; Delay to sync the tv signal
SYNC3:   DJNZ SYNC3       ; Sync it!
         LD B,C0          ; The amount of hi res lines: 192 = 8*24
GENLINE:IN A,(C)          ; Fool the ULA into thinking it is only on the
         OUT (FF),A       ; top "line" of a character by resetting the ULA ROW
                          ; counter (exactly timed with every tv VSYNC pulse)
         ADD HL,DE        ; Calculate next screen address
         CALL ULAOUT      ; and actually "jump" to it!
         DEC B            ; Decrease counter
         JP NZ GENLINE    ; Repeat until all done
         CALL 0292        ; ROM housekeeping routines - must be called!
         CALL 0220        ; ROM housekeeping routines - must be called!
         LD IX,HRESGEN    ; Reload the hi res handler address as the ROM
                          ; routines above reset it!
         JP 02A4          ; Give control back to the ROM for now
ULAOUT:  JP (HL)          ; Yes, actually, jump into the display file! -
                          ; The ULA will handle the rest - the Z80 "RET"
                          ; op-code has BIT 6 SET (C9), so will be
                          ; "executed" and control will return to the
                          ; above "calling routine" - honest!
```

- END - Pages 4 through 8, down-loaded by John Pazmino

WANTED

I would like to purchase a Keyboard 90, keyboard interface in good working condition. Please telephone or write to:

Bob Gilder, 69 Jefferson Place, Massapequa, NY 11758

Telephone 516-541-2271

Manimation

Animation for the Timex

By Donald Guess, Jr.

If you've ever tried to do computer animation in BASIC by drawing an image on the screen, erasing it, then drawing a slightly different image in its place, you know the results aren't great.

Even the most lightning-fast interpreted BASIC can't draw and erase a series of pictures fast enough to look like smooth motion. (The speed needed can range from 15–30 draws per second.) Superfast machine-code drawing routines (Apple shape-table graphics, for example) also can't move fast enough. But there are several ways to do computer animation effectively, even with BASIC. All of these methods use hardware functions to rapidly alter preformed images on the computer screen. This way, you don't have to draw each image separately while the animation is in progress.

Using character graphics—graphics that use the character set and display functions of your computer and BASIC—is one way to do it. This month's Pixel That! shows several techniques for doing character graphics animation on the new Timex 2068 computer. But even if you don't own a TS 2068, you can adapt the techniques shown here for your computer.

The first program, *Big Flap*, shows how you can do simple animation using the 2068's built-in graphics characters, plus a little string manipulation. A character array is dimensioned to contain 130 rows of five characters each. This array is then divided into 26 "frames" that are each five rows deep by five characters wide. Graphics characters are placed in these frames to look like a bat in flight.

When these frames are rapidly printed on top of each other in the same location (see lines 90–110), the effect is dramatic. The program is designed so that if you hold down any key, you can see the animation at full speed. Hands off the keyboard makes things move more slowly, letting you see each "frame" of the animation separately.

Built-in block graphics let you do some nice tricks, but they limit the amount of detail you can put in the things you animate. The 2068 offers a way around this by letting you define new character forms. This is done by bit-mapping, discussed in Pixel That! in the February 1984 issue (*see "Roving Cupid," p. 48*). Bit-mapped characters are actually tiny, fully formed pictures made up of zeros and ones of a series of binary bytes.

In the 2068, graphics characters "A–U" can be customized. Bitmaps for these characters (eight bytes per character) are contained in RAM at the address stored at system variable UDG at 23675 decimal.

Redefining graphic characters by POKEing new values into their bitmaps lets you create a small library of custom shapes. These can be displayed almost immediately in any location by a simple PRINT statement.

The second example program, *Manimate*, defines characters "A–T" in one graphic character set as successive portions of a picture of a man walking downstairs. It defines these same characters in another character set to create a man walking to the right. By PRINTing the first set of these characters to the screen in the proper position (by the string manipulation technique demonstrated in *Big Flap*), the picture of a man walking downstairs comes to life.

Then, by suddenly switching from one character set to the other (POKEing the address of the second alternative character set to the system variable UDG), the image library is instantly shifted, and a very similar code is used to animate the man walking to the right of the screen. The program is set up so you can see the animation at full speed by holding down any key. But slow it down for observation by letting up on the keyboard.

DONALD GUESS, JR. *is a freelance illustrator/cartoonist living in Columbus, Ohio. He's a member of the Association of Timex Sinclair Users (ATSU) group.*

TIMEX SINCLAIR/*BIG FLAP*

2068 • 48K RAM

```
10 DIM B$(130,5): LET C=0
20 FOR X=0 TO 11: READ A$
30 FOR Y=1 TO LEN A$
40 LET I=INT (C/5)+1: LET J=C-I*5+6
50 IF A$(Y)="R" THEN LET C=C+4: GO TO 80
60 IF A$(Y)="Q" THEN GO TO 80
70 LET B$(I,J)=CHR$ (CODE A$(Y)+63)
80 LET C=C+1: NEXT Y: NEXT X
90 FOR I=1 TO 126 STEP 5
100 IF INKEY$="" THEN PAUSE 20
110 FOR J=0 TO 4: PRINT AT 7+J,14;B$(I+J): NEXT J:
```

```
NEXT I: GO TO 90
1000 DATA "QFAKQQFAKQQQPQQRRQGQJQQFQKQQQPQQ"
1010 DATA "RRBIQECQFQKQQQPQQRRMQQQMQGQJQQQP"
1020 DATA "QQRRRDGQJDQQPQQRRRMMQMMQQPQQRRRR"
1030 DATA "DDPDDRRRRMMPMMRRRREMPMICQQQBRRRQ"
1040 DATA "MPMQJQQQGRRRQEPIQECQBICQQQBRRQQP"
1050 DATA "QQQJQGQJQQQGRRQQPQQQJQGQFQQQKRRQ"
1060 DATA "QPQQQJQGQQKQFQRRQQPQQQFQKQQFQKQR"
1070 DATA "RQMPMQQGQJQRRRQLPHAQCQBQRRQIQEQQ"
1080 DATA "LPHQRRRQOQNQQCPBARRRQPQPQAAPAARR"
1090 DATA "REHQLIQQPQQRRRBHQLCQQPQQRRQMQMQQ"
1100 DATA "FQKQAAPAARRQGQJQQFQKQQQPQQRRQFQK"
1110 DATA "QQFQKQQQPQQRRQEDIQQFQKQQQPQQRR"
```

TIMEX SINCLAIR/*MANIMATE*

2068 • 48K RAM

```
10 CLEAR 64699:LET L=29
20 DIM D$(24,3):LET D$(1)="AB":LET D$(2)="CD":LET D$(3
)="  ":LET D$(4)="EF":LET D$(5)="GH":LET D$(6)=" I":LE
T D$(7)="JK":LET D$(8)="LM":LET D$(9)=" N":LET D$(10)=
"OP":LET D$(11)="QR":LET D$(12)="ST"
30 LET D$(13)="AB":LET D$(14)="CD":LET D$(15)="EF":LET
 D$(16)="GH":LET D$(17)="IJ":LET D$(18)="KL":LET D$(19
)="  ":LET D$(20)="MN":LET D$(21)="OP":LET D$(22)="  "
:LET D$(23)="QR":LET D$(24)="ST"
40 POKE 23675,176:POKE 23676,254:FOR Z=1 TO 2:IF Z=2 T
HEN POKE 23675,5
50 FOR I=144 TO 163:FOR J=0 TO 7:READ A:POKE USR CHR$
I+J,A:NEXT J:NEXT I:NEXT Z:POKE 23675,88:POKE 23676,25
5
60 DIM E$(24,3):LET E$(1)="AB ":LET E$(2)="CD ":LET E$
(3)="EF ":LET E$(4)="GH ":LET E$(5)="IJ ":LET E$(6)="K
L ":LET E$(7)="MN ":LET E$(8)="OP ":LET E$(9)="QR ":LE
T E$(10)="ST ":LET E$(11)=" U ":LET E$(12)=" A "
70 LET E$(13)=" B ":LET E$(14)=" C ":LET E$(15)=" D ":
LET E$(16)=" E ":LET E$(17)=" FG":LET E$(18)=" HI":LET
 E$(19)=" JK":LET E$(20)=" LM":LET E$(21)=" NO":LET E$
(22)=" PQ":LET E$(23)=" RS":LET E$(24)=" TU"
80 POKE 23675,93:POKE 23676,253:FOR Z=1 TO 2:IF Z=2 TH
EN POKE 23675,181:POKE 23676,252
90 FOR I=144 TO 164:FOR J=0 TO 7:READ A:POKE USR CHR$
I+J,A:NEXT J:NEXT I:NEXT Z
100 FOR X=1 TO 24:FOR Y=1 TO 3:IF CODE D$(X,Y)<>32 THE
N LET D$(X,Y)=CHR$ (CODE D$(X,Y)+79)
110 IF CODE E$(X,Y)<>32 THEN LET E$(X,Y)=CHR$ (CODE E$
(X,Y)+79)
120 NEXT X:NEXT Y
130 POKE 23676,254:FOR I=10 TO 18:POKE 23675,176:FOR J
=1 TO 22 STEP 3:IF J=13 THEN POKE 23675,5
140 IF INKEY$="" THEN PAUSE 20
150 PRINT AT I,2;D$(J):PRINT AT I+1,2;D$(J+1):PRINT AT
 I+2,2;D$(J+2)
160 PAUSE 7:NEXT J:NEXT I
170 POKE 23675,93:POKE 23676,253:PRINT AT 19,2;E$(1):P
RINT AT 20,2;E$(2):PAUSE 10:PRINT AT 19,2;E$(3):PRINT
AT 20,2;E$(4):PAUSE 10
180 FOR I=2 TO L:POKE 23675,93:POKE 23676,253:FOR J=5
TO 24 STEP 2
190 IF INKEY$="" THEN PAUSE 20
200 PRINT AT 19,I;E$(J):IF J=11 THEN POKE 23675,181:PO
KE 23676,252
210 PRINT AT 20,I;E$(J+1):PAUSE 7:NEXT J:NEXT I
220 POKE 23675,93:POKE 23676,253:PRINT AT 19,I;E$(3):P
```

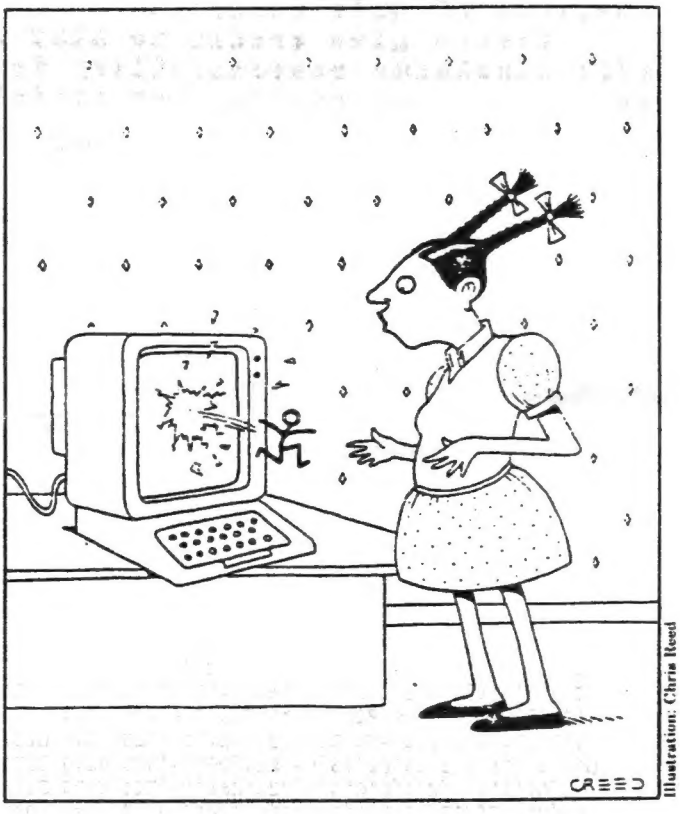

Illustration: Chris Reed

```
RINT AT 20,I;E$(4):PAUSE 5:PRINT AT 19,I;E$(1):PRINT A
T 20,I;E$(2):PAUSE 10
230 IF L=2 THEN STOP
240 LET L=L-3
250 GOTO 130
1000 DATA 0,1,2,2,1,6,10,18,0,128,64,64,128,96,80,72
1010 DATA 128,17,2,2,2,2,2,2,72,136,64,64,64,64,64,64
1020 DATA 0,0,0,1,2,2,1,6,0,0,0,128,64,64,128,96,10
1030 DATA 18,34,33,2,3,1,0,80,72,80,144,64,64,64
1040 DATA 64,64,64,0,0,0,0,0,0,0,0,0,0,1,2,2,0
1050 DATA 0,0,0,0,128,64,64,1,6,10,17,34,1,3,1
1060 DATA 128,96,80,72,80,160,64,64,64,64,64,0,0
1070 DATA 0,0,0,0,0,0,0,0,1,2,2,0,0,0,0,0,128,64
1080 DATA 64,1,6,10,18,34,33,2,3,128,96,80,72,80
1090 DATA 144,64,64,1,0,0,0,0,0,0,0,64,64,64,0,0
1100 DATA 0,0,0,0,0,0,0,0,1,2,2,0,0,0,0,0,128,64
1120 DATA 64,1,6,10,18,18,17,2,2,128,96,80,72,72
1130 DATA 136,64,64,2,2,2,0,0,0,0,0,64,64,64
1140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
1150 DATA 128,2,2,1,3,10,18,10,9,64,64,128,96,80
1160 DATA 72,68,132,2,2,2,2,2,2,0,0,64,64,128,128
1170 DATA 0,0,0,0,0,1,2,2,1,6,10,18,0,128,64,64
1180 DATA 128,96,80,72,10,5,2,2,2,2,2,0,68,130
1190 DATA 192,128,0,0,0,0,0,1,2,2,1,6,10,18,0,128
1200 DATA 64,64,128,96,80,72,10,9,2,2,2,2,2,0,68
1210 DATA 132,64,192,128,0,0,0,0,1,2,2,1,6,10,0
1220 DATA 128,64,64,128,64,96,80,10,9,1,1,1,1,1,0
1230 DATA 80,144,128,128,128,128,128,0,0,1,2,2,1
1240 DATA 3,2,6,0,128,64,64,128,64,64,64,3,3,1,1
1250 DATA 1,1,1,0,64,192,0,0,0,0,0,0,0,1,2,2,1,3
1260 DATA 3,3,0,128,64,64,128,64,64,64,2,3,1,1,1
1270 DATA 1,1,0,64,160,64,64,32,32,64,0,0,0,1,1,0
```